A Spatiotemporal Approach to Extract the 3D Trajectory of the Baseball from a Single View Video Sequence

Hubert Shum and Taku Komura
City University of Hong Kong,
hubert.shum@student.cityu.edu.hk, itaku@cityu.edu.hk

Abstract

In this paper, we propose a new method to extract and calculate the 3D trajectory of a pitched baseball in a video clip. Comparing to previous methods, which require video clips from multiple view points, only a single-view, television clip is required for our method. Since global search methods based on dynamic programming is used to find the trajectory of the ball, the system is more robust than previous incremental methods. Therefore, our technique can be used to analyze the pitches not only in live TV programs, but also in previous games by famous pitchers. It is also possible to display the 3D trajectory of the baseball in a virtual environment from the viewpoint of the hitter. Pitchers can improve their skills by viewing the trajectory of their balls, and the hitters can view the pitches of various pitchers. As a result, our system can be used for baseball training, as well as for entertainment such as video games. The method to extract the ball from the scene is also applicable to sports such as tennis, volleyball, and soccer.

1. Introduction

Baseball-related industry is huge in the world. Many people enjoy watching live matches in television during the baseball season. Players including amateurs and professionals spend a lot of time and money to train and improve their skills. They do not hesitate to pay a lot of money for good bats, gloves and balls. The best reference for the baseball industry is professional baseball. In most of the baseball video games, players in the professional league pitch, hit and run on behalf of the game players. Amateur or even professional baseball players watch video clips of famous top players to find out how they can improve their skills and become top athletes. Therefore, there is a strong demand to automatically retrieve various data from video clips.

In addition to the motion by the players, the information that many people are interested in is the trajectory of the ball. The fast balls and curve balls pitched by top athletes are quite attractive to the audience. André [2] proposed a method to extract the trajectory of a baseball from video clips using Kalman filter. Their system was created for the ESPN sports channel to give the audience better ideas whether the ball have passed the strike zone or not. By using stereo vision, the 3D trajectory of the ball was calculated. The Kalman filter approach is an incremental approach that tracks the baseball in the scene based on prediction and pattern matching. Only the information of the ball in the previous frames is used in order to predict the position of the ball in the next frame. In case there are many textures and objects in the scene which has similar color with the baseball, the system can fail to find the ball. Even though global search methods that use the information in the future frames would give better results, because of the limitation of live broadcasting, such an approach could not be chosen. A number of cameras were prepared so that even in case the tracking fails with some cameras, the 3D trajectories could be properly reconstructed.

When extracting the baseball out of the video clips, techniques such as background subtraction [3] can be used. However, such methods assume that the background of the scene does not change in the video clip. Simple subtraction fails when both the players and baseball move and overlap each other in the video. Additional methods must be combined to extract the whole trajectory of the ball from the scene.

Pingali *et al.*[4] tried to find the trajectory of the tennis ball in the tennis matches. He successfully separated the motion of the players and the ball. However, his method takes advantages of the tennis game characteristics such as the green ball color, uniform ground color, special viewing angels, etc. These characteristics cannot be applied to baseball games, because the color of the ground is not uniform, the color of the ball is often similar to the clothes of the players, and a lot of advertisements appear in the

0-7803-8603-5/04/$20.00 ©2004 IEEE.

background that makes it difficult to find out the ball in the scene.

In this paper, we propose a new global search method based on dynamic programming to extract the trajectory of the pitched baseball in TV video clips. The information of all the frames during pitching is used in order to find out the trajectory of the ball. As global search is done, the system is more robust than incremental search methods. The clips can be in non-uniform layout, low resolution and low frame rate. This guarantees the availability of video sources, and enables the analysis of old video clips. The system also reconstructs 3D baseball trajectories from single viewpoint clips. The method to extract ball from a scene can be used not only for baseball games, but also for various sports such as tennis and soccer. The system can be applied for entertainment, baseball training, and sports analysis.

2. Methodology

The input to the system is a video clip of a baseball game and the output is the 3D trajectory data of the baseball. The overview of the algorithm is shown in Figure 1. The detail of each processing is explained in the following subsections.

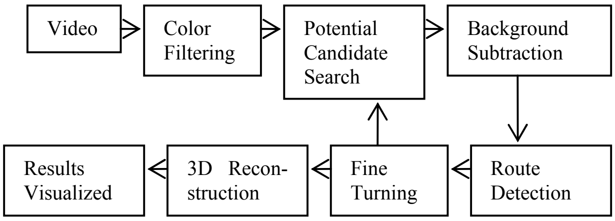

Figure 1: Overview of the methodology

2.1 Color Filtering

Every frame of the video clip is extracted and analyzed in IUV space. As the baseball has a light grey color close to white, it shows high intensity (I value) and low color distance (U and V values). As a result, it is possible to define a cylinder in IUV space which describes the color range of the baseball:

$$I > I_{Min}$$
$$\sqrt{U^2 + V^2} < Diff_{Max}$$

where I_{Min} and $Diff_{\max}$ are the thresholds which values were set to 150 and 35 respectively.

A morphological filter based on the color in this cylinder can be applied to smooth the resultant frame. Figure 2 shows the result after color filtering and morphological filtering.

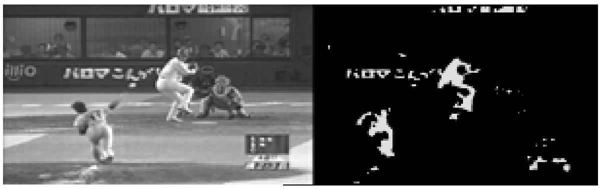

Figure 2: The original image (left), and the image after color filtering (right)

Figure 3: The potential candidates found in a shot (left). The white-on-white scenario when the ball overlapping to the player (right)

2.2 Potential Candidate Search

After the color filtering process, most of the objects are cleared out except those having a color similar to the baseball. Among the remaining objects, the system has to find out which of them is the baseball. This process is called the potential candidate search.

The constraints the potential candidates must satisfy are (1) they must be solid objects in baseball color, (2) they must be small enough to fit in the bounding box of a baseball size, and (3) they must be disconnected from any other objects in baseball color. The system will search for all potential candidates in the frame that satisfy these conditions. The potential candidates in Figure 2 are shown in Figure 3 (left).

The algorithm above will fail to trace the baseball when it passes over other white objects as shown in Figure 3 (right). This is the so-called "white-on-white" scenario. André [1] solves the problem by adjusting the camera viewing angle such that the ball will not overlap with other objects. However, this method can only be applied to pre-configured videos. In our system, the missing position is recovered in later stages.

2.3 Background Subtraction

Color and shape information are not enough to find out the correct candidate. As Figure 3 (left), there are still several potential candidates due to the complex background and noise. Background subtraction is used to select the correct candidate among them. Each potential candidate is compared to the corresponding position in the previous frame. The number of pixels having large color difference between the two frames is counted. Since the baseball is moving in high speed, such count must be significantly high for the true candidate. By background subtraction most of the potential candidates that is not the ball will be filtered.

2.4 Route Detection using Dynamic Programming

After the first three stages, an array of baseball positions in different frames is found. However, there are still chances that some of the data are not the ball. Route detection based on dynamic programming is proposed in this stage to find out the correct route among such data.

Circles in Figure 4 represent the baseball location with respect to frame number. There may be frames containing several candidates of the baseball, or containing none due to "white-on-white" scenario. Suppose there are two frames, frame number *i* and *j* *(i<j),* both containing candidates of the baseball. The 2D velocity of the ball using the position of the candidates can be calculated by

$$Speed_{i \to j} = \frac{\sqrt{(X_j - X_i)^2 + (Y_j - Y_i)^2}}{T_{i \to j}}$$

$$Direction_{i \to j} = \tan^{-1}\left(\frac{Y_j - Y_i}{X_j - X_i}\right)$$

where (X_i, Y_i) (X_j, Y_j) are the positions of the ball in frame *i* and *j,* and $T_{i \to j}$ is the time duration between frame *i* and *j*. Let us assume the candidates of the ball as nodes of a graph. When the velocity of the ball calculated by candidates in frame *i* and *j* satisfy the conditions of the pitched ball, the nodes corresponding to these candidates will be connected by an edge. As a result, we will obtain a graph as shown in Figure 4, which represents the relationship of the candidates in all the frames.

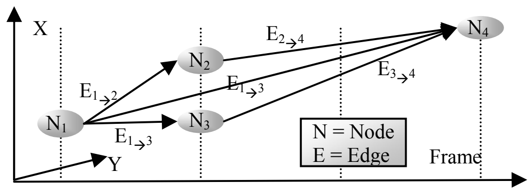

Figure 4: Potential relationship graph

After connecting the candidates by edges, a complete route that represents the trajectory of the ball is searched. Each edge of the route has to satisfy the velocity constraint throughout the path; the difference of velocities between two adjacent edges should be within a certain range. A recursive method based on dynamic programming is used to find the route that satisfies this constraint at every node during the route and returns the maximum value for the following recursively calculated criteria:

$$L(N_i) = \max_j (w(E_{i,j}) + L(N_j))$$

where *L(Ni)* represents the longest route starting from node *Ni, j* is the counter for all the edges going out from node *Ni*, and *w(Ei,j)* is the weight for edge *Ei,j* which are all set to 1. The route starting from the node with the largest *L(Ni)* can be considered as the route corresponding to the trajectory of the ball. As this method is a global search method, even though the ball is completely missed for a number of frames because of occlusion, the overall trajectory of the ball can be found.

2.5 Fine Tuning

The route now links baseball positions in different frames. Still, there is a chance that in some frames the ball is missing. We can further search for the ball in the missing frames based on the route calculated in the previous stage. First, a continuous B-spline curve interpolating the nodes of the route is generated. For frames that we could not find the baseball using the methods previously explained, we can guess the position of the ball using this curve.

The baseball is searched near the corresponding area in the curve. This time, the pixels are filtered by looser constraints so that the system has more chance to find the ball. By only applying the subtraction, it is possible to restore the position of the ball in the frames the ball could not be found previously. The chance to find out the position of the ball in the frame just after the ball is released and the frame the catcher catches the ball greatly increases this time. The chance of finding the baseball under the white-on-white scenario also increases.

2.6 3D Reconstruction

After the 2D trajectory of the ball is obtained, the 3D trajectory is calculated. This can be done by estimating the depth of the ball in the scene. The differential equation of a pitched ball can be written by the following form:

$$m\frac{\mathbf{dv}}{\mathbf{dt}} = -\frac{1}{2}c_d \rho S \|\mathbf{v}\|^2 \bullet \frac{\mathbf{v}}{\|\mathbf{v}\|} + \frac{1}{2}c_s \rho S \boldsymbol{\omega} \times \frac{\mathbf{v}}{\|\mathbf{v}\|} + m\mathbf{g}$$

where **v** is the velocity of the ball, $\boldsymbol{\omega}$ is the angular velocity of the spin, **g** is the acceleration by the gravity, *m* is the ball mass, ρ is the air density, *S* is the cross section area of the ball, and c_d, c_s are aerodynamic coefficients. The first term represents the air resistance force, the second term the Magnus force, and the third term the gravity. In order to calculate the depth of the ball, it is possible to ignore the effect of the Magnus force as the ball is moving almost perpendicular to the screen. The differential equation of the depth z can be represented here by the following equation:

$$m\frac{d^2z}{dt^2} = -\frac{1}{2}c_d\rho S\left(\frac{dz}{dt}\right)^2$$

Then, the z coordinate can be written in the following form:

$$z(t) = z_0 + \alpha\log(t + C)$$

where $\alpha = 2m / c_d\rho S$ and z_0 is the initial depth value of the ball. As the value of the parameters that compose α are all known ($m = 0.15$ $c_d = 0.3$, $\rho = 1.2$, $S = 0.0154$), the only unknown parameter is C. As the distance between the pitcher and the catcher is known (=18.44m), it is possible to calculate the value of C using the boundary conditions at the first and last frame. Finally, the depth information for all the frames is calculated.

3. Experiment

A number of video clips of different baseball games were analyzed by the system. The results are encouraging. Normally, the baseball position will be found unless it is entirely overlapped with white objects. Even though the ball cannot be found out for a number of frames, due to the global search method, the route of the ball can be found in most of the cases.

Figure 5 shows a scene of a video clip which is difficult for analysis due to white advertisement and hitter. There are fourteen frames that the baseball appears in the screen. After color filtering, candidate search and background subtraction, thirteen potential candidates, were found through the animation (Figure 5), five of them are actually not the baseball. Route detection was used to find the seven correct candidates among the twelve (Figure 6, left). After fine tuning, nine baseball locations were found. (Figure 6, right) All of them are correct positions. The 3D trajectory can be generated as shown in Figure 7.

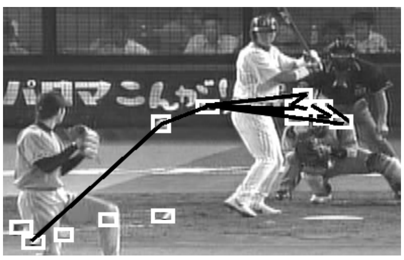

Figure 5: The potential candidates presented by white boxes linked by black edges.

Figure 6: The list of balls found after the recursive route search (left) and those found after fine tuning (right)

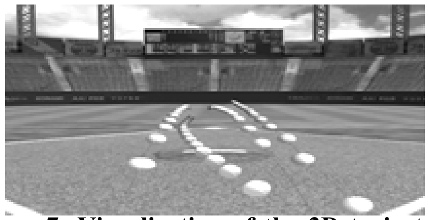

Figure 7: Visualization of the 3D trajectories of various pitches.

4. Conclusion

In this paper, we proposed a new global search method to extract the 2D trajectory of a pitched baseball using dynamic programming. Based on the 2D trajectory and aerodynamics, the 3D trajectory of the ball was reconstructed. Because the pitched ball passes over backgrounds including advertisement and the hitter, simple pattern matching and motion tracking techniques are not enough to extract the ball. Instead, a method based on graph search was introduced. Through experiments, it was shown that the method provides high successful rate of recognition.

5. References

[1] André Guéziec, "Tracking a Baseball for Broadcast Television", *ACM Siggrapph 2003 Courses*.

[2] D. Li, "Moving Objects Detection by Block Comparison", *The 7th IEEE International Conference on Electronics, Circuits and Systems*, pp. 341-344.

[3] D. B. Brown, "Motion-Based Foreground Segmentation".

[4] G. S. Pingali, Y. Jean, and I. Carlbom, "Real Time Tracking for Enhanced Tennis Broadcasts", *Proceedings IEEE Computer Vision and Pattern Recognition*, pp. 260-265.